UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

AM

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/058,647	01/28/2002	David M. Ungar	004-4556	3845

22120 7590 06/17/2005

ZAGORIN O'BRIEN GRAHAM LLP
7600B N. CAPITAL OF TEXAS HWY.
SUITE 350
AUSTIN, TX 78731

EXAMINER	
INGBERG, TODD D	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 06/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/058,647	UNGAR, DAVID M.	
	Examiner	Art Unit	
	Todd Ingberg	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 23 December 2004.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-26 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-26 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

Claims 1 – 26 have been examined.

Claim 1 has been amended.

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 1 – 10, 12 – 23 and 26 are rejected under 35 U.S.C. 101 for being non statutory.

The examiner has provided one means of overcoming this rejection.

Claim 1
A multipass parser **implementation product executing on a computer and stored on a computer readable medium** comprising: plural miniparsers each successively operable on a respective abstract syntax tree that corresponds to an input information encoding and that includes transformations of predecessor ones, if any, of the miniparsers, wherein respective ones of the miniparsers are limited to particular subsets of syntactic constructs to be parsed in the input information encoding.

Claim 13
A method **executing on a computer and stored on a computer readable medium** of implementing a parser for an input information encoding, the method comprising: defining a succession of miniparsers each operable on a respective parse state resulting from a predecessor one of the mimparsers, wherein each of the miniparsers recognizes only a particular subset of syntactic constructs to be parsed in the input information encoding.

Claim 23
A method **executing on a computer and stored on a computer readable medium** of parsing an information encoding, the method comprising: performing plural successive transformations, each successive one of the transformations operating on an abstract syntax tree that is a result of a predecessor one of the transformations, each of the successive transformations handling only a subset of syntactic constructs to be recognized in the information encoding.

Claim 26
An apparatus comprising: encoded information **on a computer readable medium and executing on a computer** substantially in accordance with a grammar; multipass means for performing plural successive transformations on the encoded information, each successive transformation handling only a subset of syntactic constructs in accordance with the grammar.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1 – 26 are rejected under 35 U.S.C. 102(b) as anticipated by or, in the alternative, under 35 U.S.C. 103(a) as obvious over the combination of "Support for Modular Parsing in Software Reengineering", by Peake et al, published in IEEE July 1997 in view of the theory of parsing and lexical analysis that one of ordinary skill in the art should know as taught by Aho et al, in Compilers Principles Tools and Techniques from September 19, 1985.

Knowledge of the Ordinary Artisan in the Art

5. The following is knowledge one of very ordinary skill in the art should know prior to invention. The college text book Compilers Principles, Techniques and Tools, by Aho et al. published September 12, 1985 (Red Dragon Book) contains the basic principles of parser theory.

Interpretation

6. Claims 4, 5, 18 and 26 and contain the word "***substantially***", in order to provide a non indefinite meaning the terms are interpreted to mean. The code meets syntax and semantic requirements of a language, which is inherent in the art.

A. ***abstract syntax tree*** – Aho pages 2, 6- 7, 28-30,40-43, 49, 160, 169-171, 196, 279, 287 – 290 and 296
B. ***transformations of predecessor*** -
C. ***syntactic constructs*** – Aho, page 13
D. ***input information encoding*** – Aho page 13, input to each step.
E. ***multipass parser*** – Aho pages 20 - 22

F. ***output of a respective predecessor*** – Aho building Nodes and leafs of AST above.
G. ***input and output abstract syntax trees are separately encoded*** - Aho page 13, definition of steps.
H. ***syntactic constructs are those of a programming language*** – Aho, page 13.
I. ***information encoding includes code substantially*** - The code meets syntax and semantic requirements of a language
J. ***syntactic constructs are those defined by a grammar*** – Aho, page 13
K. ***grammar*** – Aho page 25
L. ***comment parser*** – Aho white space and comments, pages 54, 84-85, 99
M. ***a delimiter parser*** – Aho page 28 semicolon
N. ***a top-level statement parser*** – Aho pages 48 – 52, adapting the transformation scheme
N. ***a compilation unit parser*** -
O. ***a name parser*** – Aho, Symbol table building page 56, identifiers and keywords
P. ***hierarchy of functional transformations*** - Processing of ASTs above
Q. ***parse state*** – Aho , pages 104 - 105
R. ***parse tree*** – AST above
S. ***lexer*** – Aho, pages 58, 105 to 113
T. ***non-comment tokens*** – Code other than comment parser above.
NOTE: Page 13 of Aho shows the relationship from input grammar through the parsing/lexical analysis process of input to output of different steps.
Motivation to Combine

Peake teaches a modular parser with multi-parsers. What Peake does not explicitly teach is the underlying theory of parse theory. It is Aho who teaches the theory of parsing . One of ordinary skill in the art must understand the theory of parsing in order to implement a parser. Therefore, it would be obvious to one of ordinary skill in the art to combine the teachings of Peake and Aho because, "Parsing is the process of determining if a string of tokens can be generated by a grammar." (Aho, page 40, section 2.4).

Claim 1
Peake teaches a multipass product (**Peake**, page 62, figure 3, use of generic routines using the signatures require multiple passes to parse the components of language statements) parser (**Peake**, page 58, Introduction) implementation (**Peake,** page 64, 5.4 through 6.1) comprising: plural miniparsers (**Peake**, page 59, last paragraph, parsers) each successively operable (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers) on a respective abstract syntax tree that corresponds to an input information encoding and that includes transformations of predecessor ones (**Aho**, pages 2, 6- 7, 28-30,40-43, 49, 160, 169-171, 196, 279, 287 – 290 and 296), if any, of the miniparsers, wherein respective ones of the miniparsers are limited to particular subsets of syntactic constructs to be parsed in the

input information encoding (**Peake**, page 60 section 3.3 signature used in generic routines to determine input encoding).

Claim 2
The multipass parser implementation of claim 1, wherein, for at least some of the miniparsers (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers), the respective abstract syntax tree is an output of a respective predecessor one of the miniparsers (Aho, ASTs as per claim 1 and (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers).

Claim 3
The multipass parser implementation of claim 1, wherein, for at least some of the miniparsers (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers), respective input and output abstract syntax trees are separately encoded(**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers)..

Claim 4
The multipass parser implementation of claim 1, wherein the syntactic constructs are those of a programming language; and wherein the information encoding includes code substantially in accordance therewith. (**Peake**, page 59, last full paragraph "Through the use of….").

Claim 5
The multipass parser implementation of claim 1, wherein operation of each of the miniparsers is on a substantial entirety of its respective abstract syntax tree. In view of claim 1 and the inherency of ASTs to parse theory as taught by Aho.

Claim 6
The multipass parser implementation of claim 1, wherein the syntactic constructs are those defined by a grammar; but wherein none of the miniparsers individually implements the grammar (**Peake**, page 60, section 2).

Claim 7
The multipass parser implementation of claim 1, wherein the syntactic constructs are those defined by a grammar; and wherein the operation of successive ones of the miniparsers corresponds to the grammar(**Peake**, page 60, section 2).

Claim 8
The multipass parser implementation of claim 1, wherein the plural miniparsers (As per claim 1) include: a comment parser (Aho, page 54); a delimiter parser (**Peake,** page 60, section 2, grammar by definition the corresponding relationship of the language input and parser to make tokens); a top-level statement parser (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers) ; a compilation unit parser; and a name parser (**Peake,** page 63, section 4.4 – namespace names of routines and variables and constants in symbol table).

Claim 9
The multipass parser implementation of claim 1, wherein the plural miniparsers correspond to an hierarchy of functional transformations. (**Aho,** page 5, 2. Hierarchical analysis and the collective meaning as a result).

Claim 10
The multipass parser implementation of claim 1, embodied as a computer program product encoded in at least one computer readable medium (**Peake**, as per claim 1 and page 65, section 6.3, compile time and run time requires computer program product encoding).

Claim 11
The multipass parser implementation of claim 10, wherein the at least one computer readable medium is selected from the set of a disk (**Peake**, page 65, compile and runtime and CASE tool are stored on disk),, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium.
NOTE : Examiner selected to meet the disk in view of the limitations "OR ".

Claim 12
A software engineering tool including the plural miniparsers of claim 1 (**Peake**, as per claim 1 and page 65, CASE tools and section 6.3 compiler).

Claim 13
Peake teaches a method of implementing a parser for an input information encoding(**Peake**, page 59, last full paragraph "Through the use of…."), the method comprising: defining a succession of miniparsers (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers) each operable on a respective parse state resulting from a predecessor one of the miniparsers (**Peake**, page 64, section 5.3), wherein each of the miniparsers recognizes only a particular subset of syntactic constructs to be parsed in the input information encoding (**Peake**, page 59, last sentence second to last paragraph on left recognisor and scheme on right side of page several bullet items combinator parsing).

Claim 14
The method of claim 13, wherein operation of each successive miniparser transforms an output of a predecessor miniparser (**Peake**, page 59, scheme on right side of page several bullet items combinator parsing).

Claim 15
The method of claim 13, wherein operation of each successive miniparser successively (as per claim 13) refines a parse tree that corresponds to the input information encoding (**Peake**, page 59, last full paragraph "Through the use of….").

Claim 16
The method of claim 13, wherein operation of each successive miniparser (as per claim 13) produces a distinct parse tree (inherent as per claim 1) that represents a successive refinement

corresponding to the input information encoding (**Peake**, page 59, last full paragraph "Through the use of....").

Claim 17
The method of claim 13, further comprising: executing the miniparsers in succession (**Peake**, page 64, section 5.4).

Claim 18
The method of claim 13, wherein the syntactic constructs are those of a programming language; and wherein the input information encoding includes code substantially in accordance therewith. (**Peake**, page 59, last full paragraph "Through the use of....").

Claim 19
The method of claim 13, wherein a first executed one of the miniparsers implements a lexer. Inherent as taught by Aho pages 58 and 105 to 113.

Claim 20
The method of claim 13, wherein a first executed one of the miniparsers operates on a lexed encoding corresponding to the input information. Inherent as taught by Aho pages 58 and 105 to 113. Also see page 13 of Aho to see the input output relationship.

Claim 21
The method of claim 13, applied to a compilation unit of program code, wherein an earlier executed one of the miniparsers (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers) associates comment tokens of a compilation unit with respective non-comment tokens (**Peake**, teaches building of tokens page 61 and the grammar being extensible page 61, section 4 page 65, section 7.2); and wherein a later executed one of the miniparsers matches grouping tokens (**Peake**, page 60, last sentence the sequence of tokens). **Interpretation:** Associates being comment on same line as code.

Claim 22
The method of claim 21, wherein a still later executed one of the miniparsers segregates tokens into toplevel statements (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers).

Claim 23
A method of parsing an information encoding, the method comprising: performing plural successive transformations(**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers), each successive one of the transformations operating on an abstract syntax tree that is a result of a predecessor one of the transformations (**Aho** as per claim 1), each of the successive transformations handling only a subset of syntactic constructs to be recognized in the information encoding. (**Peake**, page 59, last full paragraph "Through the use of...." and **Peake**, page 59, last sentence second to last paragraph on left recognisor and scheme on right side of page several bullet items combinator parsing).

Claim 24
Peake teaches a computer program product encoded in at least one computer readable medium (Peake, page 65, compile and runtime and CASE tool)and comprising: functional encodings of at least two miniparsers(**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers), a first one of the miniparsers executable to transform a first parse tree into a second parse tree and the second one of the miniparsers executable to transform the second parse tree into a third parse tree (**Aho,** parse trees are inherent in parsing); each of the at least two miniparsers recognizing only a subset of syntactic constructs to be parsed in an information encoding (**Peake**, page 59, last sentence second to last paragraph on left recognisor and scheme on right side of page several bullet items combinator parsing)to which the first, second and third parse trees correspond (**Aho** , as per claim 1).

Claim 25
The computer program product of claim 24, wherein the at least one computer readable medium is selected from the set of a disk (Peake, page 65, compile and runtime and CASE tool are stored on disk), tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium.
NOTE : Examiner selected to meet the disk in view of the limitations "OR ".

Claim 26
Peake teaches an apparatus comprising: encoded information substantially in accordance with a grammar (**Peake**, page 60, section 2); multipass (As per claim 1)means for performing plural successive transformations on the encoded information, each successive transformation handling only a subset of syntactic constructs in accordance with the grammar (**Peake**, page 61, above 3.3 High level parser combinators combine parsers to create new parsers).

Response to Arguments

Applicant's arguments of December 23, 2004 have been considered but are not persuasive.

Rejection Under 35 USC §101

Applicant's argument is directed toward a "Safe Harbor". Not toward the fact that the original and amended claims are not claiming the invention which is software as stored on a tangible device (computer readable medium). The invention is executing on a computer so the one proposed way of overcoming the rejection which mentions executing is redundant.

Rejection Under 35 USC §102 and §103

Applicant's argument's support the Examiner's decision to reject the Application under both 35 U.S.C §102 and 35 U.S.C §103. Applicant's is arguing inherent features of parsing theory. On page 7 the Applicant states that the Peake reference, "does not consider issues of parse tree construction." The Examiner's response is that Peake should not bore a reader of ordinary skill with inherent structures. Parse trees are inherent in parsers. The Examiner made of record in FAOM, a 1956 reference from Noam Chomsky of MIT, the title is "Three Models For Description of Languages". He invented programming language theory as we know it. Please,

revisit this article to see the inherent parse trees. You can't have a parse without the construction of parse trees.

The Aho reference is a well know text book which discloses the mundane details of parse theory. It does not disclose Modular Parser. It is Peake who teaches modular parsers. That is why the rejection under 35 U.S.C §103 was provided. The attack on Aho is piecemeal.

Applicant continues to state "... neither Peak or Aho teach or suggest miniparsers that that successive operations operate on abstract syntax trees or parse states resulting form a predecessor miniparser." First, Peakes modular parsers are miniparsers. Second, successive operations are not an option. The programming statement must be processed and ASTs must be constructed. To get from the form of programming language statement to an AST requires successive operations. Aho covers the inherent successive operations by displaying before and after of the lexical analysis on pages 5 – 8 (overview and Chapter 2 provides a detail teaching. The result of a parser as disclosed in the Aho reference is covered in the sections mentioned just prior. To say Peake does not produce a concatenated result is to ask the Peake's article is proposing Modular parsing that will not result in the required result of a parse operation. The Examiner has not located a passage in Peake that says the concept of a modular parser is a great idea but will not result in the required result of a parser.

Applicant states" ... Although Peake discloses implementing modular parsers for specific grammars, it does not limit parsers to particular subsets of syntactic constructs". Examiner's response is that this is a good thing. The AST's are available to the debugger as the passage quoted by Applicant. The debugger uses the ASt information for it's symbol table. This should be well known to one of ordinary skill in the art. Not limiting the parsers to particular subsets of syntactic constructs is also good. This means the parser is not hard coded and can be reused for different language syntax. Peake also states another advantage in the cited passage on pages 7 – 8.

Applicant's argument directed toward claim 1 in view of the Examiner's response above should clarify the inherent aspects of parse theory which meet most of the arguments present. As for the example of LISP being shown. The specific language syntax does not distinguish the claimed invention.

Applicant's arguments for claims 2 and 3 support the Examiner's position that the successive operations is met by the Peake reference and that parse trees are inherent. Aho in it's teaching of the different steps of a parser show how the input and output are separately encoded. Although, Aho does not teach modular parsers, the input and output of different stage is shown. In a similar manner when one of ordinary skill in the art reviews Peake and focuses on the functional of a modular parser the different components (modules) as they relate to the syntax of the input must be separately encoded. It would make no sense to parse the same construct in another part of the parser.

As for claim 6, the argument that Peake teaches more than the disclosed invention is not grounds for patentablility.

As for claim 8, the Examiner only finds an allegation and no technical argument.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (571) 272-3723. The examiner can normally be reached on during the work week..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Todd Ingberg
Primary Examiner
Art Unit 2193

TI